An Approach to Online Identification of Takagi-Sugeno Fuzzy Models

Plamen P. Angelov, *Member, IEEE,* and Dimitar P. Filev, *Senior Member, IEEE*

***Abstract*—An approach to the online learning of Takagi–Sugeno (TS) type models is proposed in the paper. It is based on a novel learning algorithm that recursively updates TS model structure and parameters by combining supervised and unsupervised learning. The rule-base and parameters of the TS model continually evolve by adding new rules with more summarization power and by modifying existing rules and parameters. In this way, the rule-base structure is inherited and up-dated when new data become available. By applying this learning concept to the TS model we arrive at a new type adaptive model called the Evolving Takagi–Sugeno model (ETS). The adaptive nature of these evolving TS models in combination with the highly transparent and compact form of fuzzy rules makes them a promising candidate for online modeling and control of complex processes, competitive to neural networks. The approach has been tested on data from an air-conditioning installation serving a real building. The results illustrate the viability and efficiency of the approach. The proposed concept, however, has significantly wider implications in a number of fields, including adaptive nonlinear control, fault detection and diagnostics, performance analysis, forecasting, knowledge extraction, robotics, behavior modeling.**

***Index Terms*—Online recursive identification, rule-base adaptation, Takagi–Sugeno models.**

I. Introduction

TAKAGI–SUGENO models have recently become a powerful practical engineering tool for modeling and control of complex systems. They form a natural transition between conventional and rule-based control by expanding and generalizing the well-known concept of gain scheduling. While the gain-scheduling [24] paradigm is based on the assumption of local approximation of a nonlinear system by a collection of linear models, the TS models utilize the idea of linearization in a fuzzily defined region of the state space. Due to the fuzzy regions, the nonlinear system is decomposed into a multi-model structure consisting of linear models that are not necessarily independent [4].

The TS model representation often provides efficient and computationally attractive solutions to a wide range of control problems introducing a powerful multiple model structure that is capable to approximate nonlinear dynamics, multiple operating modes and significant parameter and structure variations.

Manuscript received May 24, 2002; revised December 5, 2002. This work used data generated from the ASHRAE Project RP1020. This paper was recommended by Associate Editor L. O. Hall.

P. P. Angelov is with the Department of Communications Systems, Lancaster University, Bailrigg, Lancaster LA1 4YR, U.K. (e-mail: p.angelov@lancaster.ac.uk).

D. P. Filev is with the Ford Motor Co., Detroit, MI 48239 USA (e-mail: dfilev@ford.com).

Digital Object Identifier 10.1109/TSMCB.2003.817053

The methods for learning TS models from data are based on the idea of consecutive structure and parameter identification [3], [14]. Structure identification includes estimation of the focal points of the rules (antecedent parameters) by fuzzy clustering. With fixed antecedent parameters, the TS model transforms into a linear model. Parameters of the linear models associated with each of the rule antecedents are obtained by pseudo-inversion or by applying the recursive least square (RLS) method [16], [28]. Alternatively, the antecedent parameters can be considered as initial estimates only and the structure and parameters can be further optimized by back-propagation [20] or genetic algorithm [12]. These methods, however, suppose that *all* the data is available at the start of the process of training. Therefore, they are appropriate for *offline* applications only. Their use in *online* algorithms is only possible for the price of re-training the whole model structure and parameters with iterative and time-consuming procedures such as back-propagation [5], genetic algorithms [6]–[8], [10]–[12] or other nonlinear search techniques [1], [21], [29].

Although some objects, including biotechnological processes, building thermal systems, etc. have relatively slow dynamics, making such re-training possible, it is difficult to characterize it as *adaptation*, especially in respect to the model *structure*. It is in fact, a procedure where completely new models are repeatedly generated given the new data. The fact that fuzzy models are still not adaptive, while in many practical problems the control object or the environment is changing significantly is an important obstacle in their design, which is still unresolved [19].

For continuous *online* learning of the TS models a development of a new *online* clustering method responsible for model structure (rule base) learning *online* is needed. This requires recursive calculation of the informative potential of the data [9], which represents a spatial proximity measure used to define the focal points of the rules (antecedent parameters). If suppose that the model *structure evolves* similarly to the model parameters, though much slower, then we need suitable new algorithms for *online* clustering and recursive parameter estimation with this assumption. The purpose of this paper is to present such algorithms and the results of their application to a number of test cases both simulated and real.

Recently, rule-bases [9], [15], [27] and neural networks [20], [31] with *evolving* structure have been developed. Rule-base of Initial Conditions, RBIC [15], Intelligent Model Bank [27] and the Self-constructing fuzzy-neural network controller [20] are primarily oriented to control applications. They use different mechanism of rules update based on the distance to certain rule center [15], [27], [31] or the error in previous steps [20].

Evolving rule-based ($e\mathrm{R}$) models [9] use the informative potential of the new data sample (accumulated spatial proximity information) as a trigger to update the rule-base, which ensures greater generality of the structural changes. Outliers have no chance to become rule centers. It also ensures that the rules are more general (that they are able to describe a larger number of data samples) from time of their initialization. In addition, the mechanism of rule-base modification (replacement of a less informative rule with a more informative one) is considered in [9]. It is also based on the informative potential and is more conservative than the replacement used in [15], [27], [31] ensuring a *gradual* change of the rule-base structure and inheritance of the structural information. $e\mathrm{R}$ models generate a new rule if there is significant new information present in the data collected. The evolution mechanism takes care of the replacement of existing rules based on the accumulated measure of the spatial proximity of *all* the data samples. If the informative potential of the new data sample is higher than the average potential of the existing rules it is added to the rule base. If the new data, which is accepted as a focal point of a new rule is too close to a previously existing rule then the old rule is replaced by the new one.

The appearance of a new rule indicates a region of the data space that has not been covered by the initial training data. This could be a new operating mode of the plant or reaction to a new disturbance. In reality, many regimes and process states cannot be practically included into the training data set (such as faulty process behavior), but states close to them could well appear during the process run [22].

It is important to note that learning could start without a priori information and only one data sample. This interesting feature makes the approach potentially very useful in adaptive control, robotic, diagnostic systems and as a tool for knowledge acquisition from data.

The concept of $e\mathrm{R}$ modeling [9] is further developed here in respect to *online* identification of ETS models. *Recursive* procedures for calculation of the informative potential of the new data and of the consequence parameters are introduced, which remove the need of the time moving window considered in [9]. This feature is vitally important for *real-time* applications.

The rest of the paper is organized as follows. The problem of identification of TS models is presented in Section II. Two alternative ways (globally and locally optimal) of calculation of the consequent parameters are presented. The new approach for *online* learning ETS models is presented in the next Section III. In Section IV the essential stages of the procedure are defined and systematically described. Section V studies experimental results considering a real air-conditioning engineering problem. Concluding remarks are given in Section VI.

II. TS Fuzzy Model and the Problem of its Identification

Fuzzy model identification has its roots in the pioneering papers of Sugeno and his coworkers [13], [14] and is associated with the so-called Takagi-Sugeno (TS) fuzzy models—a special group of rule-based models with fuzzy antecedents and functional consequents that follow from the Takagi–Sugeno–Kang reasoning method:

$$\mathfrak{R}_i: \boldsymbol{IF}(x_1 \; is \; \aleph_{i1}) \; \boldsymbol{AND} \ldots \boldsymbol{AND}(x_n \; is \; \aleph_{in}) \; \boldsymbol{THEN}(y_i = a_{i0} + a_{i1}x_1 + \ldots + a_{in}x_n); \quad i = \{1, R\} \quad (1)$$

where $\mathfrak{R}_i$ denotes the i^{th} fuzzy rule; R is the number of fuzzy rules; $\boldsymbol{x}$ is the input vector; $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$; $\aleph_{ij}$ denotes the antecedent fuzzy sets, $j = \{1, n\}$; y_i is the output of the i^{th} linear subsystem; a_{il} are its parameters, $l = \{0, n\}$.

The TS model paradigm [13] can be considered as a generalization of the gain-scheduling concept. Instead of linearizing strictly at an operating point it utilizes the idea of linearization in a fuzzily defined region of the space. The fuzzy regions are parameterized and each region is associated with a linear subsystem. Owing to the fuzzily defined antecedents, the nonlinear system forms a collection of loosely coupled multiple linear models. The degree of firing of each rule is proportional to the level of contribution of the corresponding linear model to the overall output of the TS model. For Gaussian-like antecedent fuzzy sets

$$\boldsymbol{\mu}_{ij} = \boldsymbol{e}^{-\alpha\left\|x_j - x_{ij}^*\right\|^2}; \quad i = \{1, R\} \quad j = \{1, n\} \qquad (2)$$

where $\alpha = 4/r^2$ and r is a positive constant, which defines the spread of the antecedent and the zone of influence of the i^{th} model (radius of the neighborhood of a data point); too large a value of r leads to averaging, too small a value—to over-fitting; values of $\mathrm{r} \in [0.3; 0.5]$ can be recommended) [16]; x_i^* is the focal point of the i^{th} rule antecedent.

The firing level of the rules are defined as Cartesian product or conjunction of respective fuzzy sets for this rule

$$\tau_i = \mu_{i1}(x_1) \times \mu_{i2}(x_2) \times \ldots \times \mu_{in}(x_n) = \bigcap_{j=1}^{n} \mu_{ij}(x_j). \quad (3)$$

The TS model output is calculated by weighted averaging of individual rules' contributions

$$y = \sum_{i=1}^{R} \lambda_i y_i = \sum_{i=1}^{R} \lambda_i x_e^T \pi_i \qquad (4)$$

where $\lambda_i = (\tau_i / \sum_{j=1}^{R} \tau_j)$ is the normalized firing level of the i^{th} rule; y_i represents the output of the i^{th} linear model; $\pi_i = [a_{i0}\ a_{i1}\ a_{i2} \ldots a_{in}]^T$, $i = [1, R]$, is the vector of parameters of the i^{th} linear model; $x_e = \left[1\ x^T\right]^T$ is the expanded data vector.

Generally, the problem of identification of a TS model is divided into two sub-tasks [3], [13], [16].

i) Learning the antecedent part of the model (1), which consists of determination of the focal points of the rules, i.e., the centers (x_i^*; $i = [1, R]$) and spreads (r) of the membership functions.
ii) Learning the parameters of the linear subsystems (a_{ij}; $i = [1, R]$; $j = [0, n]$) of the consequents.

A. Learning Rule Antecedents by Data Space Clustering

First sub-task can be solved by clustering the input-output data space ($z = [x^T; y]^T$). The Subtractive Clustering method [16], Fuzzy C-means [17], and the Gustafson–Kessel clustering

method [23] are among the well-established methods for learning the antecedent parameters *offline* in a batch-processing learning mode when all the input-output data is available.

The procedure called *subtractive clustering* [16] is an improved version of the so-called *mountain clustering* approach [25]. It uses the data points as candidate prototype cluster centers. The capability of a point to be a cluster center is evaluated through its potential—a measure of the spatial proximity between a particular point z_i and *all other* data points

$$P_i = \frac{1}{TD}\sum_{j=1}^{TD} e^{-\alpha\|z_i - z_j\|^2}; \quad i = [1, TD] \tag{5}$$

where P_i denote the potential of the i^{th} data point and where TD is the number of training data).

As seen from (5) the value of the potential is higher for a data point that is surrounded by a large number of close data points. Therefore, it is reasonable to establish such a point to be the center of a cluster [24]. The potential of all other data points is reduced by an amount proportional to the potential of the chosen point and inversely proportional to the distance to this center. The next center is found also as the data point with the highest (after this subtraction) potential. The procedure is repeated until the potential of all data points is reduced below a certain threshold.

The procedure of the *subtractive clustering* includes the following steps [16].

1) Initially, the data point with the highest potential is chosen to be the first cluster center

$$P_1^* := \max_{i=1}^{TD} P_i \tag{6}$$

where P_1^* denotes the potential of the first center.

2) The potential of all other points are then reduced by an amount proportional to the potential of the chosen point and inversely proportional to the distance to this center

$$P_i := P_i - P_k^* e^{-\beta\|z_i - z_k^*\|^2} \quad i = [1, N] \tag{7}$$

where P_k^* denotes the potential of the k^{th} center; $\mathrm{k} = [1, \mathrm{TD}]$; $\beta = (4/r_b^2)$; where r_b is a positive constant, determining the radius of the neighborhood that will have measurable reductions in the potential because of the closeness to an existing center; recommended value of r_b is $r_b = 1.5r$ [16].

3) Two boundary conditions are defined: lower $(\underline{\varepsilon}^* P^{ref})$ and upper $(\overline{\varepsilon}^* P^{ref})$ threshold, determined as a function of the maximal potential called the "reference" potential (P^{ref}). A data point is chosen to be a new cluster center, and respectively center of a rule, if its potential is higher than the upper threshold.

4) If the potential of a point lies between the two boundaries, the shortest of the distances (δ_{min}) between the new candidate to be a cluster center (z_k^*) and all previously found cluster centers is decisive. The following inequality, express the trade-off between the potential value and the closeness to the previous centers

$$\frac{\delta_{\min}}{r} + \frac{P_k^*}{P_1^*} \geq 1. \tag{8}$$

This approach has been used for initial estimation of the antecedent parameters in fuzzy identification. It relies on the idea that each cluster center is representative of a characteristic behavior of the system [16]. The resulting cluster centers are used as parameters of the antecedent parts defining the focal points of the rules of the model.

B. Learning Parameters of Linear Subsystems

For fixed antecedent parameters the second sub-task, estimation of the parameters of the consequent linear models can be transformed into a least squared problem [2]. This is accomplished by eliminating the summation operation in (4) and replacing it with an equivalent vector expression of y

$$y = \psi^T \theta \tag{9}$$

where $\theta = [\pi_1^T, \pi_2^T, \ldots \pi_R^T]^T$ is a vector composed of the linear model parameters; $\psi = [\lambda_1 x_e^T, \lambda_2 x_e^T, \ldots, \lambda_R x_e^T]^T$ is a vector of the inputs that are weighted by the normalized firing levels of the rules.

For a given set of input-output data (x_k^T, y_k), $k = [1, TD]$, the vector of linear model parameters θ minimizing the objective function is

$$J_G = \sum_{k=1}^{TD} \left(y_k - \psi_k^T \theta\right)^2 \tag{10}$$

where $\psi_k = [\lambda_1(x_k) x_{ek}^T, \lambda_2(x_k) x_{ek}^T, \ldots, \lambda_R(x_k) x_{ek}^T]^T$; $x_{ek} = [1, x_k^T]^T$, can be estimated by the recursive least squares algorithm (called also the Kalman filter) [13], [16]

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_k \left(y_k - \psi_k^T \hat{\theta}_{k-1}\right) \tag{11}$$

$$C_k = C_{k-1} - \frac{C_{k-1}\psi_k \psi_k^T C_{k-1}}{1 + \psi_k^T C_{k-1} \psi_k}; \quad k = [1, TD] \tag{12}$$

with initial conditions $\hat{\theta}_0 = 0$ and $C_0 = \Omega I$, where Ω is a large positive number; C is a $R(n+1) \times R(n+1)$ co-variance matrix; $\hat{\theta}_k$ is an estimation of the parameters based on k data samples.

Alternatively, the objective function (10) can be written in vector form as

$$J_G = (Y - \Psi^T \theta)^T (Y - \Psi^T \theta) \tag{10a}$$

where the matrix Ψ and vector Y are formed by ψ_k^T, and y_k, $k = [1, TD]$.

Then the vector θ minimizing (10a) could be obtained by the pseudo-inversion

$$\theta = (\Psi^T \Psi)^{-1} \Psi^T Y. \tag{13}$$

The objective functions (10), (10a) are globally optimal, but this does not guarantee locally adequate behavior of the sub-models that form the TS model [18]. Locally meaningful sub-models

could be found using the locally weighted objective function [18], [28]

$$J_L = \sum_{i=1}^{R} (Y - X^T\pi_i)^T \Lambda_i (Y - X^T\pi_i) \tag{14}$$

where matrix X is formed by x_{ek}^T; $X \in R^{TD\times(n+1)}$; matrix Λ_i is a diagonal matrix with $\lambda_i(x_k)$ as its elements in the main diagonal.

An approximate solution minimizing the cost function (14) can be obtained by assuming the linear subsystems are loosely coupled with levels of interaction expressed by the weights $\lambda_i(x_k)$. Then (14) can be regarded as a sum of cost functions

$$J_L = \sum_{i=1}^{R} J_{Li}$$

where

$$J_{Li} = (Y - X^T\pi_i)^T \Lambda_i (Y - X^T\pi_i). \tag{15}$$

The solutions π_i that minimize the weighted least square problems expressed by the objective functions J_{Li} can be obtained by applying a weighted pseudo-inversion [18], [28]

$$\pi_i = (X^T\Lambda_i X)^{-1} X^T \Lambda_i Y \quad i = [1, R]. \tag{16}$$

Alternatively, a set of solutions to individual cost functions J_{Li} (vectors π_i's) can be recursively calculated through the weighted RLS (wRLS) algorithm. In this case, a wRLS algorithm that minimizes each of the cost functions J_{Li} is applied to the linear subsystem associated with *each* rule (see the Appendix for the detailed derivation)

$$\hat{\pi}_{ik} = \hat{\pi}_{ik-1} + c_{ik}x_{ek}\lambda_i(x_k)\left(y_k - x_{ek}^T\hat{\pi}_{ik-1}\right) \tag{17}$$

$$c_{ik} = c_{ik-1} - \frac{\lambda_i(x_k)c_{ik-1}x_{ek}x_{ek}^T c_{ik-1}}{1 + \lambda_i(x_k)x_{ek}^T c_{ik-1}x_{ek}}; \quad k = [1, TD] \tag{18}$$

with initial conditions $\hat{\pi}_0 = 0$ and $c_{i0} = \Omega I$.

As seen from (17), (18) when the normalized firing weight of certain rule $\boldsymbol{i^*}$ is equal to 1 the wRLS algorithm transforms into RLS (11), (12) based on this rule only ($R = \{\boldsymbol{i^*}\}$). For the rule ($\boldsymbol{i^o}$) for which the normalized firing level is 0 for a certain time step $\boldsymbol{k_0}(\lambda_{10}(x_{k0}) = 0)$ the parameters and the co-variance matrix stay unchanged ($\hat{\pi}_{\text{ik}} = \hat{\pi}_{\text{ik}-1}$; $c_{\text{ik}} = c_{\text{ik}-1}$). When $0 < \lambda_1(x_k) < 1$ the update of the co-variance matrix and parameters are weighted by the normalized firing level.

III. *Online* Learning of TS Models

In *Online* mode, the training data are collected continuously, rather than being a fixed set. Some of the new data reinforce and confirm the information contained in the previous data. Other data, however, bring new information, which could indicate a change in operating conditions, development of a fault or simply a more significant change in the dynamic of the process [9]. They may posses enough new information *to form a new rule or to modify* an existing one. The value of the information they bring is closely related to the information the data collected so far already possesses. The judgement of the informative potential and importance of the data is made based on their spatial proximity, which corresponds to operating conditions, possibly seasonal variations or different faults.

online learning of ETS models includes *online* clustering under assumption of a gradual change of the rule-base and modified (weighted) recursive least squares. Due to the *evolution* of the model structure, the number of fuzzy rules is expected to grow. This is, however, significantly slower than the growth of the size of the data vectors, because the potential is inversely proportional to the number of the data points (5).

A. Online Clustering Approach

The *online* clustering procedure starts with the first data point established as the focal point of the first cluster. Its coordinates are used to form the antecedent part of the fuzzy rule (1) using for example Gaussian membership functions (2). Any other type of membership functions could also be used instead. Its potential is assumed equal to 1.

Starting from the next data point onwards the potential of the new data points is calculated *recursively*. As a measure of the potential, we use a Cauchy type function of first order

$$P_k(z_k) = \frac{1}{1 + \frac{1}{(k-1)}\sum_{l=1}^{k-1}\sum_{j=1}^{n+1}\left(d_{lk}^j\right)^2}; \quad k = 2, 3, \ldots \tag{19}$$

where $P_k(z_k)$ denotes the potential of the data point (z_k) calculated at time k; $d_{lk}^j = z_l^j - z_k^j$, denotes projection of the distance between two data points (z_l^j and z_k^j) on the axis z^j (x^j for $j = 1, 2, \ldots, n$ and on the axis y for $j = n+1$).

This function is monotonic and inversely proportional to the distance and enables *recursive* calculation, which is important for *online* implementation of the learning algorithm. Additionally, we do not subtract a specified amount from the highest potential, but update all the potentials after a new data point is available *online*.

Potential of the *new* data sample is *recursively* calculated as follows (see the Appendix for details)

$$P_k(z_k) = \frac{k-1}{(k-1)(\vartheta_k + 1) + \sigma_k - 2v_k} \tag{20}$$

where $\vartheta_k = \sum_{j=1}^{n+1}(z_k^j)^2$; $\sigma_k = \sum_{l=1}^{k-1}\sum_{j=1}^{n+1}(z_l^j)^2$; $v_k = \sum_{j=1}^{n+1} z_k^j\beta_k^j$; $\beta_k^j = \sum_{l=1}^{k-1} z_l^j$.

Parameters ϑ_k and v_k in (20) are calculated from the current data point z_k, while β_k^j and σ_k are recursively updated as $\sigma_k = \sigma_{k-1} + \sum_{j=1}^{n+1}(z_{k-1}^j)^2$; $\beta_k^j = \beta_{k-1}^j + z_{k-1}^j$.

After the *new* data are available in *online* mode, they influence the potentials of the centers of the clusters ($z_l^*, l = [1, R]$), which are respective to the focal points of the existing rules ($x_l^*, l = [1, R]$). The reason is that by definition the potential depends on the distance to ***all*** data points, including the new ones (the sum in the denominator by $\boldsymbol{l}$ in (19) has an increasing number of components). The *recursive* formula for update of the

potentials of the focal points of the existing clusters can easily be derived from (19) (see the Appendix for details)

$$P_k(z_l^*) = \frac{(k-1)P_{k-1}(z_l^*)}{k-2+P_{k-1}(z_l^*)+P_{k-1}(z_l^*)\sum_{j=1}^{n+1}\left(d_{k(k-1)}^j\right)^2} \tag{21}$$

where $P_k(z_l^*)$ is the potential at time $\boldsymbol{k}$ of the cluster center, which is a prototype of the l^{th} rule. Potentials of the *new* data points are compared to the updated potential of the centers of the existing clusters.

If the potential of the *new* data point is *higher* than the potential of the *existing* centers **then** the *new* data point is ***accepted*** as a new center and a ***new rule*** is formed with a focal point based on the projection of this center on the axis $\boldsymbol{x}$ ($R := R+1$; $x_R^* = x_k$). The rationale is that in this case the new data point is more descriptive, has more summarization power than *all* the other data points. It should be noted that the condition to have higher potential is a very strong one. The reason is that with the growing number of data, their concentration is usually decreasing except in the cases some new important region of data space reflecting a new operating regime [4] or new condition appears. In such cases a new rule is formed, while outlying data are automatically rejected because their potential is significantly lower due to their distance from the other data. This property of the proposed approach is very promising for fault detection problems.

If in addition to the previous condition (the potential of the new data point is higher that the potential of all the previously existing centers) the *new* data point is ***close to an old center***

$$\frac{P_k(Z_k)}{\max\limits_{l=1}^{R} P_k(z_i^*)} - \frac{\delta_{\min}}{r} \geq 1 \tag{22}$$

then the *new* data point (z_k) replaces this center $(z_j^* := z_k)$. This mechanism for rule-base adaptation called *modification* ensures a replacement of a rule with another one built around the projection of the new data point on the axis $\boldsymbol{x}$.

It should be noted that using the potential instead of the distance to a certain rule center only [15], [27], [31] for forming the rule-base results in rules that are more informative and a more compact rule-base. The reason is that the spatial information and history are not ignored, but are part of the decision whether to upgrade or modify the rule-base.

The proposed *online* clustering approach ensures an *evolving* rule-base by dynamically upgrading and modifying it while inheriting the bulk of the rules ($R-1$ of the rules are preserved even when a modification or an upgrade take place), Fig. 1.

B. Online Recursive Estimation of Consequence Parameters of ETS

The problem of increasing size of the training data is handled by RLS (11), (12) for the globally optimal case and wRLS (17), (18) for the locally optimal case. They, however, are based on the assumption of a constant/unchanged rule base (fixed antecedent parameters). Under this assumption, the optimization problems (10), (10a) and (14) are linear in parameters. In ETS, however, the rule-base is assumed to *be gradually evolving*. Therefore, the number of rules as well as the parameters of the antecedent part

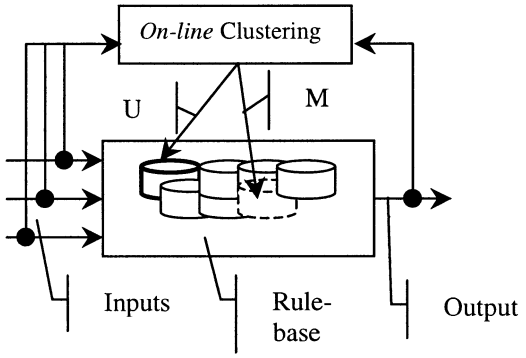

Fig. 1. Schematic representation of the rule-base evolution based on the data samples potential (M—modification/replacement); U—up-grade of a rule).

will vary, though the changes are normally significantly more rare than the time step (the change in the data set vector).

Because of this evolution, the normalized firing strengths of the rules (λ_i) will change. Since this effects *all* the data (including the data collected before time of the change) the straightforward application of the RLS (11), (12) or wRLS (17), (18) is not correct. A proper resetting of the co-variance matrices and parameters initialization of the Kalman filter (RLS) is needed at each time a rule is added to and/or removed from the rule base [2].

We propose to estimate the co-variance matrices and parameters of the new $(R+1)^{th}$ rule as a weighted average of the respective co-variance and parameters of the remaining R rules. This is possible, since the approach of rule-base innovation, we consider concerns one rule only the other R rules remain unchanged.

1) Global Parameter Estimation: The ETS model is used for *online* prediction of the output based on the past inputs

$$\hat{y}_{k+1} = \psi_k^T \hat{\theta}_k \quad k = 2, 3, \ldots \tag{23}$$

The following Kalman filter procedure is applied

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_{k-1}\left(y_k - \psi_{k-1}^T \hat{\theta}_{k-1}\right) \quad k = 2, 3, \ldots \tag{24}$$

$$C_k = C_{k-1} - \frac{C_{k-1}\psi_{k-1}\psi_{k-1}^T C_{k-1}}{1+\psi_{k-1}^T C_{k-1}\psi_{k-1}} \tag{25}$$

with initial conditions

$$\hat{\theta}_1 = \left[\hat{\pi}_1^T, \hat{\pi}_2^T, \ldots, \hat{\pi}_R^T\right]^T = 0; \quad C_1 = \Omega I. \tag{26}$$

When a new rule is added to the rule-base, the Kalman filter is reset in the following way.

i) Parameters *of the new rule* are determined by the weighted average of the parameters of the other rules. The weights are the normalized firing levels of the existing rules λ_i. The idea is to use the existing centers as a rule-base to approximate the initialization of the parameters of the new rule by a weighted sum. Parameters of the other rules are *inherited* from the previous step

$$\hat{\theta}_k = \left[\hat{\pi}_{1(k-1)}^T, \hat{\pi}_{2(k-1)}^T, \ldots, \hat{\pi}_{R(k-1)}^T, \hat{\pi}_{(R+1)k}^T\right]^T \tag{27}$$

where

$$\hat{\pi}_{R+1k} = \sum_{i=1}^{R} \lambda_i \hat{\pi}_{ik-1}. \tag{27a}$$

ii) Co-variance matrices are reset as

$$C_k = \begin{bmatrix} \rho\varsigma_{11} & \cdots & \rho\varsigma_{1R(n+1)} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho\varsigma_{R(n+1)1} & \cdots & \rho\varsigma_{R(n+1)R(n+1)} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \Omega & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & \Omega \end{bmatrix} \tag{28}$$

where ς_{ij} is an element of the co-variance matrix ($i = [1, R\times(n+1)]; j = [1, R\times(n+1)]$); $\rho = ((R^2+1)/R^2)$ is a coefficient.

In this way, the part of the co-variance matrix associated with the new $(R+1)^{th}$ rule (last $\mathrm{n}+1$ columns and last $\mathrm{n}+1$ rows) is initialized as usual (with a large number (Ω) in its main diagonal and co-variance matrices respective for the rest of the rules (from 1 to R) are updated by multiplication of ρ (28). The rationale for this is that the correction the co-variance matrices needs, to approximate *the role the new*, $(R+1)^{\mathrm{th}}$ *rule* ***would have*** if it was in the rule-base from the beginning, can be represented by ρ (see the Appendix).

When a rule is replaced by another one, which has antecedent parameter close to the rule being replaced, then parameters and co-variance matrices are inherited from the previous time step.

2) Local Parameter Estimation: The local parameter estimation is based on the wRLS

$$\hat{\pi}_{ik} = \hat{\pi}_{ik-1} + c_{ik} x_{ek-1} \lambda_i(x_{k-1}) \left(y_k - x_{ek-1}^T \hat{\pi}_{ik-1}\right); \quad k = 2, 3, \ldots \tag{29}$$

$$c_{ik} = c_{ik-1} - \frac{\lambda_i(x_{k-1}) c_{ik-1} x_{ek-1} x_{ek-1}^T c_{ik-1}}{1 + \lambda_i(x_{k-1}) x_{ek-1}^T c_{ik-1} x_{ek-1}}; \quad i = [1, R] \tag{30}$$

with initial conditions

$$\hat{\pi}_1 = 0 \text{ and } c_{i1} = \Omega I. \tag{31}$$

In this case, the co-variance matrices are separate for each rule and have smaller dimensions ($c_{ik} \in R^{n+1\times(n+1)}$; $i = 1[1, R]$). Parameters of the newly added rule are determined as weighted average of the parameters of the rest R rules by (27a). Parameters of the other R rules are inherited ($\pi_{ik} := \pi_{i(k-1)}$; $i = [1, R]$).

When a rule is replaced by another rule, which have close antecedent parameter (center) then parameters of all rules are inherited ($\pi_{ik} := \pi_{i(k-1)}$; $i = [1, R]$).

The co-variance matrix of the newly added rule is initialized by

$$c_{R+1k} = \Omega I. \tag{32}$$

The co-variance matrices of the rest R rules are inherited ($c_{ik} := c_{i(k-1)}$; $i = [1, R]$).

IV. Procedure for Rule-Base Evolution in ETS Models

The *recursive* procedure for *online* learning of ETS models, introduced in this paper, includes the following stages.

1) Stage 1: Initialization of the rule-base structure (antecedent part of the rules).
2) Stage 2: At the next time step reading the *next* data sample.
3) Stage 3: *Recursive* calculation of the potential of each new data sample to influence the *structure* of the rule-base.
4) Stage 4: *Recursive* up-date of the potentials of old centers taking into account the influence of the *new* data sample.
5) Stage 5: Possible ***modification*** or up-grade *of* the rule-base *structure* based on the potential of the new data sample in comparison to the potential of the *existing* rules' centers (focal points).
6) Stage 6: *Recursive* calculation of the consequent parameters.
7) Stage 7: Prediction of the output for the next time step by the ETS model.

The execution of the algorithm continues for the next time step from stage 2. It should be noted that the first output to be predicted is $\hat{y}_3$.

Stage 1. The rule-base could contain one single rule only, based, for example, on the first data sample. Then

$$k := 1; \quad R := 1; \quad x_1^* := x_k; \quad P_1(z_1^*) := 1; \quad \theta_1 = \pi_1 = 0; \quad C_1 = \Omega I \tag{33}$$

where z_1^* is the first cluster center; x_1^* is focal point of the first rule being a projection of z_1^* on the axis $\boldsymbol{x}$.

In principle, the rule-base could be initialized by existing expert knowledge. Generally, however, it could be based on the *off-line* identification approaches, described in Section II. In this case

$$R := R^{ini}; \quad P_1(z_i^*) := 1; \quad i = [1, R^{ini}] \tag{34}$$

where R^{ini} denotes the number of rules defined initially *off-line*.

Stages 2 to 7 are performed *online*. They form the distinctive characteristics of the proposed approach.

Stage 2. At the next time step $(k := k+1)$ the *new* data sample (z_k) is collected.

At **stage 3** the potential of each new data sample is *recursively* calculated by (20). The use of already calculated values ϑ_k and β_k^j leads to significant time and calculation savings because (19) is normally calculated from large matrices (the number of training data in *online* mode is continuously growing). At the same time, they have accumulated information regarding the spatial proximity of *all* previous data.

At **stage 4** the potentials of the focal points (centers) of the existing clusters/rules are *recursively* updated by (21).

At **stage 5** the potential of the *new* data sample is compared to the updated potential of existing centers and a decision whether to *modify* or up-grade the rule-base is taken.

```
a)
  IF (the potential new data point
  is higher than the potential of the
```

existing centers: $P_k(z_k) > P_k(z_I^*);$ $i = [1, R]$)
AND [the new data point is *close to an old center* (22)]
THEN *the new data point* (z_k) *replaces it.*

In this case, the *new* data point is used as a prototype of a focal point (let us suppose that it has index j

$$z_j^* = \arg\min_{i=1}^{R} \|z_k - z_i^*\|) : x_j^* := x_k; \quad P_k(z_j^*) := P_k(z_k). \tag{35}$$

Consequence parameters and co-variance matrices are *inherited* from the rule to be replaced

$$\hat{\pi}_k := \hat{\pi}_k^{j^*}; \quad C_k := C_k^{j^*} \tag{36}$$

It should be noted that when a rule is replaced by another rule the weights (λ) are changing according to (4) and the summation in the denominator in (4) should change. $R-1$ addends in this summation are the same and *only one change*. Moreover, since the new center is close to the replaced one by definition (22), this change is marginal. The disturbance caused to the RLS by this change could be ignored, because the Kalman filter is able to cope with this disturbance starting from the existing estimations of the parameters and co-variance matrices. This is also illustrated by the experimental results (next section).

b)
ELSE IF (the potential of the new data point is *higher* than the potential of the *existing* centers: $P_k(z_k) > P_k(z_I^*);\ i = [1, R]$)
THEN *it is added to the rule-base as a new rule's center.*

In this case, the new data point becomes a prototype of a focal point of a *new* rule

$$R := R + 1; \quad x_R^* = x_k; \quad P_k(z_R^*) = P_k(z_k). \tag{37}$$

Consequence parameters and co-variance matrices are reset by (27)–(28) or (32), respectively, for the global or local estimation.

END IF

At **Stage 6** parameters of the consequence are *recursively* updated by RLS (24), (25) with initializations (26) for globally optimal parameters or by wRLS (29), (30) with initializations (31) for locally optimal parameters.

In the first case the cost function (10) is minimized, which guarantees globally optimal values of the parameters, while in the second case the locally weighted cost function (15) is minimized and locally meaningful parameters are obtained.

At **Stage 7** the output for the next time step $(k+1)$ is predicted by (23).

The algorithm continues from stage 2 by reading the next data sample at the next time step.

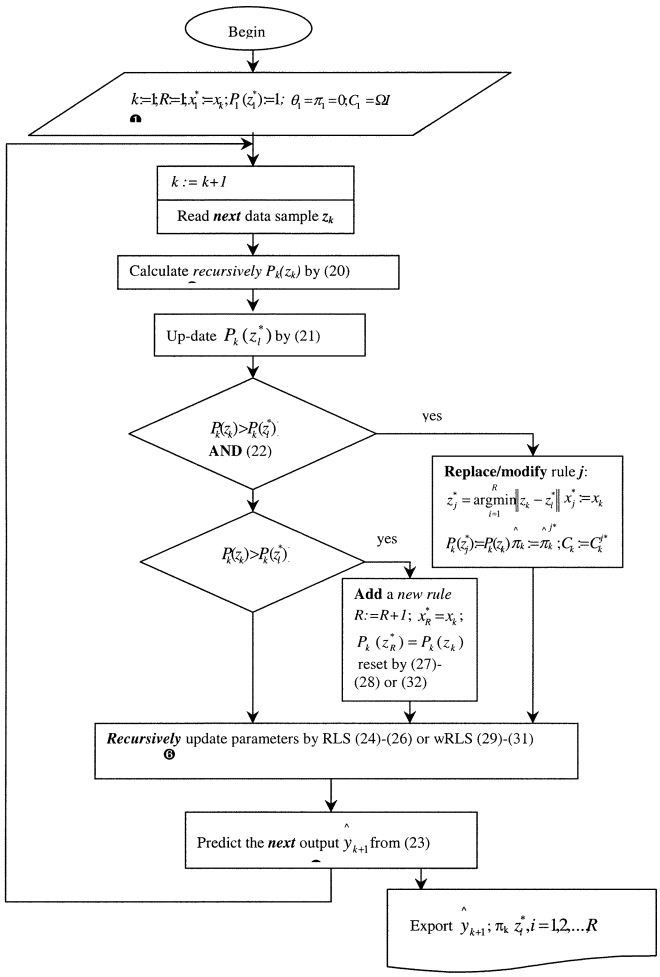

Fig. 2. Block-diagram of the *online* identification of ETS models.

A graphical representation of the algorithm that realizes the proposed approach is demonstrated in Fig. 2. All steps are *non-iterative*.

Using the approach, a transparent, compact and accurate model can be found by rule base evolution based on experimental data with the simultaneous *recursive* estimation of the fuzzy set parameters. It is interesting to note that the rate of upgrade with new rules does not lead to an excessively large rule base in comparison to [15], [20], [27], [31]. The reason for this is that the condition for the new data point to have higher potential (19), (20) than the focal points of rules of *all* existing rules is a hard requirement. Additionally, the possible proximity of a candidate center to the already existing focal points leads to just a replacement of the existing focal point, i.e. modification of its coordinates without enlarging the rule-base size.

V. Experimental Results

The new algorithm has been tested on the data from a fan-coil sub-system of an air-conditioning system serving a real building. Training data were collected on August 3, and August 19, 1998 (courtesy of ASHRAE for the use of data, generated from the ASHRAE funded research project RP1020).

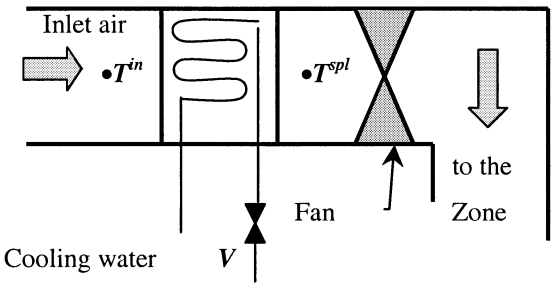

Fig. 3. Experimental set-up.

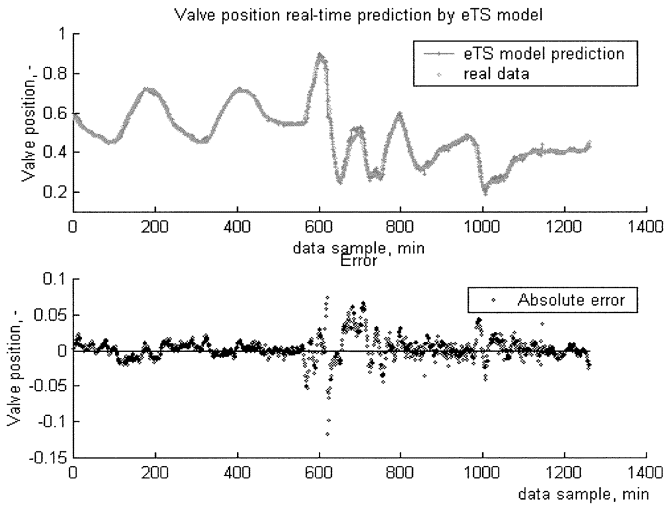

Fig. 4. Absolute error in prediction the valve position using global parameter estimation.

The ETS model of the position (V) of the valve controlling the water flow rate to a fan-coil sub-system has been considered (Fig. 3). The model makes *online* prediction of the valve position $\Delta K = 7$ steps ahead (the step in this realization is one minute). The present value of V is one of the inputs of the model considered, while the other inputs are the present and past (one step back) values of the air inlet (T^{in}) and supply to the zone (T^{spl}) temperature

$$y_{k+1} = V_{k+\Delta K}; \quad k = 1, 2, \ldots, 1263 \tag{38}$$

$$x_k = \left\{V_k; V_{k-1}; T_k^{in}; T_{k-1}^{in}; T_k^{spl}; T_{k-1}^{spl}\right\}. \tag{39}$$

The coil cools the warm air that flows on. The cool air is used to maintain comfortable conditions in an occupied Zone. One of the principle loads on the coil is generated due to the supply of ambient air required to maintain a minimum standard of indoor air quality.

The results of the *online* modeling using the global identification criteria (10) are shown in Fig. 4.

The model *evolves* to five rules and, respectively, five linear sub-models with six parameters each. The centers of the membership functions describing the fuzzy sets of the antecedent part of the rules are tabulated in Table I.

The RMS error is 0.015 59 and calculations take a fraction of a second for each new data point. The parameters are estimated in real time by the RLS (24), (25). Their evolution is depicted in Fig. 5

TABLE I
FUZZY SETS OF THE ANTECEDENTS

	V_k	V_{k-1}	T_k^{in}	T_{k-1}^{in}	T_k^{suppl}	T_{k-1}^{suppl}
x_1^*	0.647	0.640	19.58	19.58	12.44	12.394
x_2^*	0.517	0.523	18.89	18.89	13.22	13.223
x_3^*	0.696	0.693	20.31	20.39	12.57	12.573
x_4^*	0.402	0.402	27.67	27.55	12.77	12.506
x_5^*	0.405	0.405	24.56	24.51	12.81	12.863

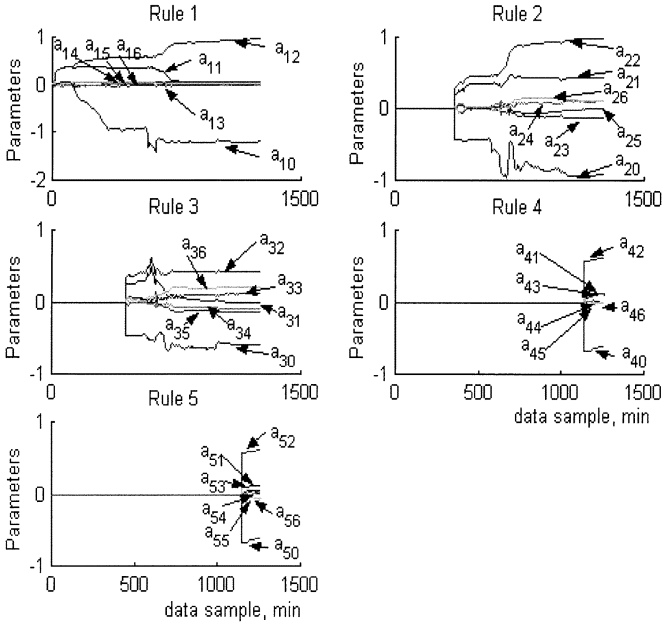

Fig. 5. Evolution of parameters of the linear sub-models (global estimation).

It is interesting to note that in time instants of adding new rule the changes of the parameters by (27), (27a) are not drastic (Fig. 5). More significant sudden changes occur in the norm of the co-variance matrix because of the resetting (28) seen from Fig. 6.

The ETS model has evolved to the same 5 rules when the local identification (14) is applied. In addition, the RMS error is marginally higher (0.016 04) (see Figs. 7–9).

The evolution of the parameters in this case is smoother and the parameters are locally more transparent.

A similar problem of modeling temperature difference across the cooling coil (Fig. 3) has been considered. The following measurements have been used:

1) flow rate of the air entering the coil $(m, kg/s)$;
2) moisture content of the air entering the coil $(g, -)$;
3) temperature of the chilled water $(T^{in}, ^\circ)$;
4) control signal to the valve $(U, -)$.

The temperature difference (drop) across the coil (δT) is predicted in *real time*

$$y_{k+1} = \delta T_{k+1} \tag{40}$$

$$x_k = \{U_k; m_k; T_k^w; g_k\}. \tag{41}$$

The data is from a full-scale air-conditioning test facility and cover two months (May and August) over two seasons (summer and spring). The data was collected with the system operating

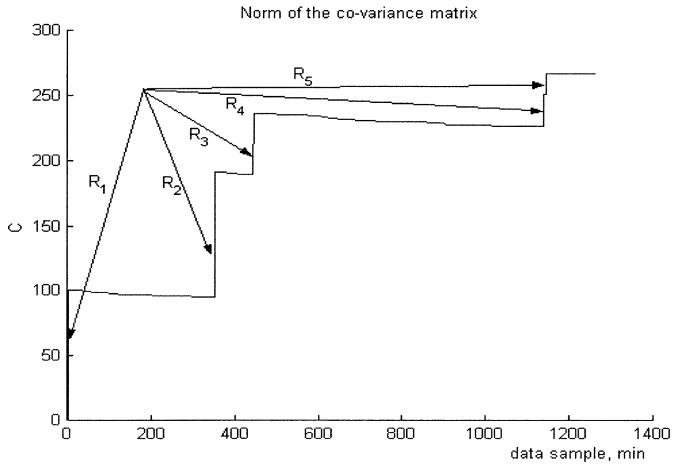

Fig. 6. Evolution of the norm of the co-variance (global estimation).

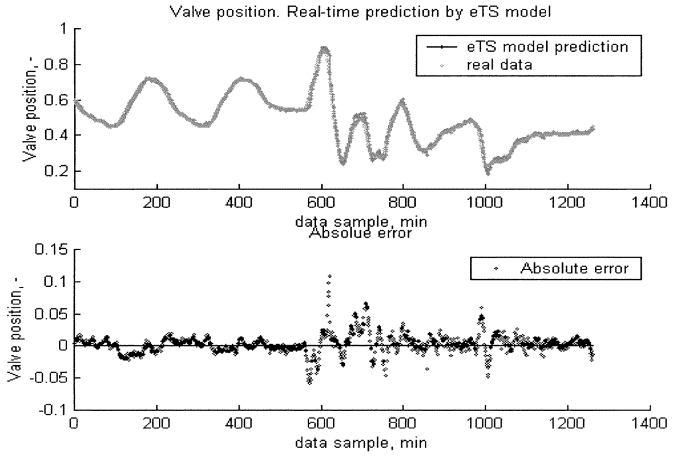

Fig. 7. Absolute error in prediction of the valve position using locally optimal linear models.

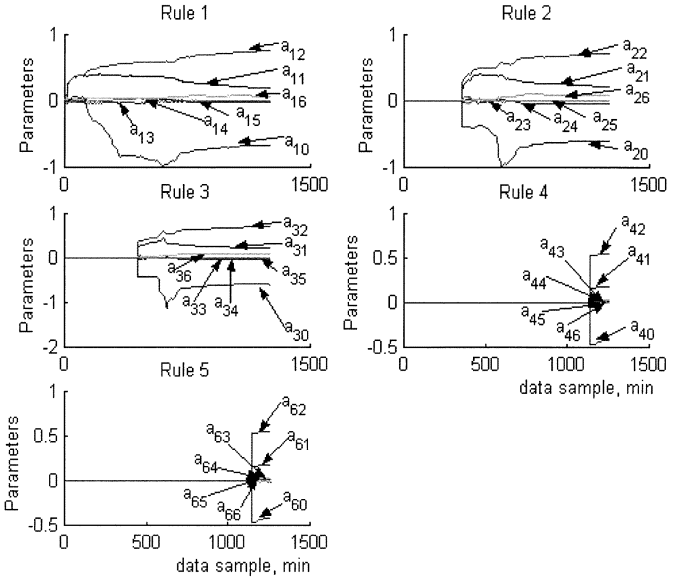

Fig. 8. Evolution of parameters of the linear sub-models (local estimation).

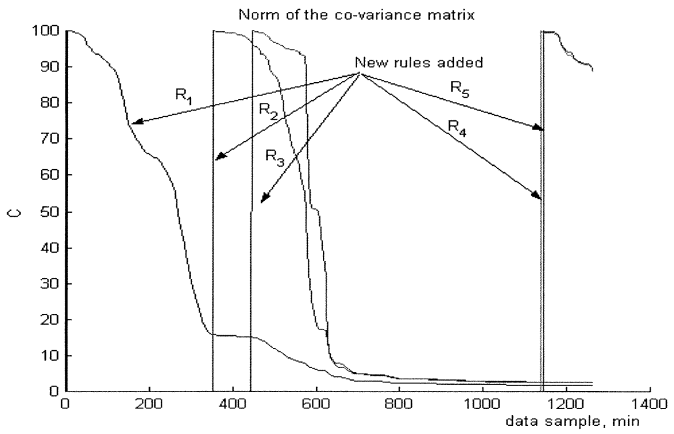

Fig. 9. Evolution of the norm of the co-variance matrices (local estimation).

under normal conditions on days in August and May respectively.

The proposed approach demonstrates that it is possible to build ETS model *online* from data of one season (summer) and then successfully to use this model making *gradual* changes to its structure and parameters for another season (spring). The RMS error is about half a degree centigrade (0.522 74 °C; nondimensional error index is 0.091 85). The model upgrades its structure to four rules with a *gradual* evolution.

The results are similar for the global and local estimation. The RMS error in local estimation is 0.656 57 °C [Fig. 10(b)]. The centers of the antecedent part at the end of the estimation are tabulated in Table II (they are the same for both type of estimation).

The robustness of the eTS model has been tested by considering 25% additive normally distributed random noise to the control signal to the valve, flow rate of the air entering the coil, and the moisture content of the air entering the coil. The error in the prediction of the temperature difference across the coil was higher, but in the same order of magnitude ($\mathrm{RMSE} = 0.6527\,^{\circ}\mathrm{C}$; $\mathrm{NDEI} = 0.1694$). As it is seen in Fig. 10(c) there are high frequency components in the output prediction, because the output from the eTS model is a function of the inputs, most of which have been noisy. One can find a more smooth prediction by proper tuning of the radius of influence of the clusters [parameter r in (2)]. It can significantly decrease the effect of the noise—in a noisy environment a larger radius of influence will prevent the algorithm from creating new clustering centers due to noise. In a limiting case in a very noisy environment the algorithm will end up with just a few cluster centers and vice versa. This is illustrated by the result, which have been achieved by increasing the value of the radius in this experiment from 0.3 to 0.5. It resulted in reduction of the RMSE to 0.6213 ($\mathrm{NDEI} = 0.1615$). A further reduction of the radius to 0.8 lead to $\mathrm{RMSE} = 0.5976$; $\mathrm{NDEI} = 0.1546$ and only three rules. We believe that a straightforward upgrade of the proposed algorithm can be obtained by adding a set of application specific rules automatically adjusting the radius of influence to the noise level.

It can be mentioned that the cluster centers are weighted by the frequencies (the number data points belonging with a high degree of membership to the particular cluster). This practically excludes the chance of an outlier to become a cluster center and to influence the output of the model. In addition, the cluster centers with low potential are periodically replaced. These intrinsic

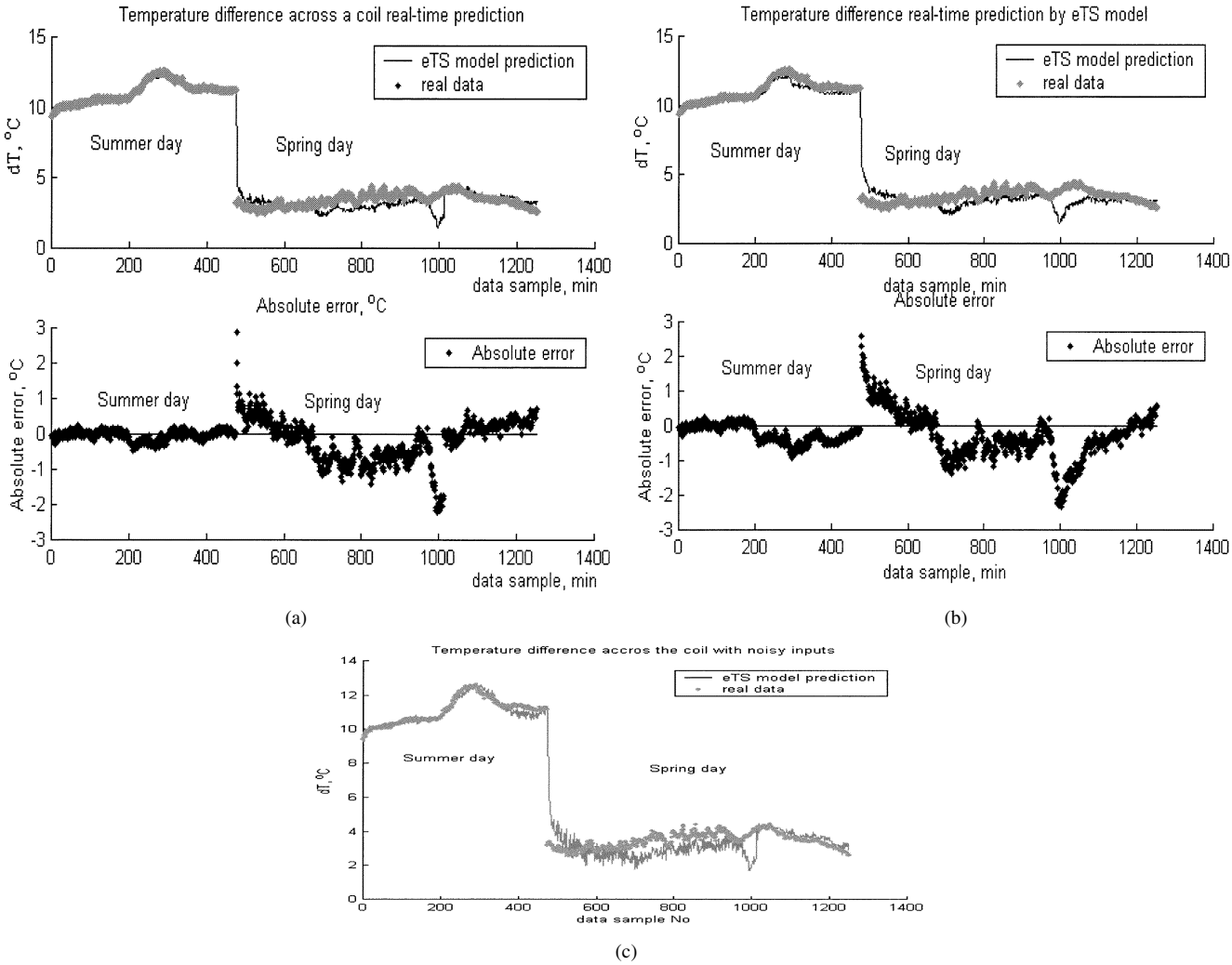

Fig. 10 (a) Prediction of the temperature difference across a coil (global criteria). (b) Prediction of the temperature difference across a coil (local criteria). (c) Prediction of the temperature difference across a coil in the presence of random noise.

TABLE II
CENTERS OF THE ANTECEDENT

	U_k-	m_k, kg/s	T_k^w, °C	g_k-
x_1^*	0.3811	0.8600	9.0894	0.0107
x_2^*	0.8464	09897	10.260	0.0101
x_3^*	0.5906	0.9870	9.8567	0.0100
x_4^*	1.0000	0.5239	12.885	0.0055

mechanisms of the ETS model design acts as a safeguard to the noisy data, which is illustrated in this example (see Figs. 11 and 12).

The proposed approach has been tested on a benchmark problem: the Mackey–Glass chaotic time series prediction and the results compared to those generated by alternative techniques for online learning TS models, published in references [31] and [34]. The chaotic time series is generated from the Mackey–Glass differential delay equation defined by [16], [31]

$$x(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t).$$

The aim is using the past values of x to predict some future value of x. We assume $x(0) = 1.2$, $\tau = 17$ and the value of the signal 85 steps ahead $x(t+85)$ is predicted (same as in [31]) based on the values of the signal at the current moment, 6, 12, and 18 steps back

$$\text{Output} = [\text{x(t} + 85)];$$
$$\text{Inputs} = [\text{x(t} - 18); \text{x(t} - 12); \text{x(t} - 6); \text{x(t)}].$$

The validation data set consists of 500 data samples. The same nondimensional error index (NDEI) defined as the ratio of the root mean square error over the standard deviation of the target data is used as in [31]–[34] to compare model performance.

The results summarized in Table III and Fig. 13 show that the new approach can yield a compact model with favorably comparable NDEI. It should be noted that TS models with lower NDEI have been reported in [31]–[34]. The number of rules (nodes or units) of these models is, however, in the range of a thousand which significantly undermines their transparency an interpretability. ETS model has evolved in *online* mode to 113 transparent rules with $\text{NDEI} = 0.0954$. It should also be noted that similar approach to the one presented in this paper,

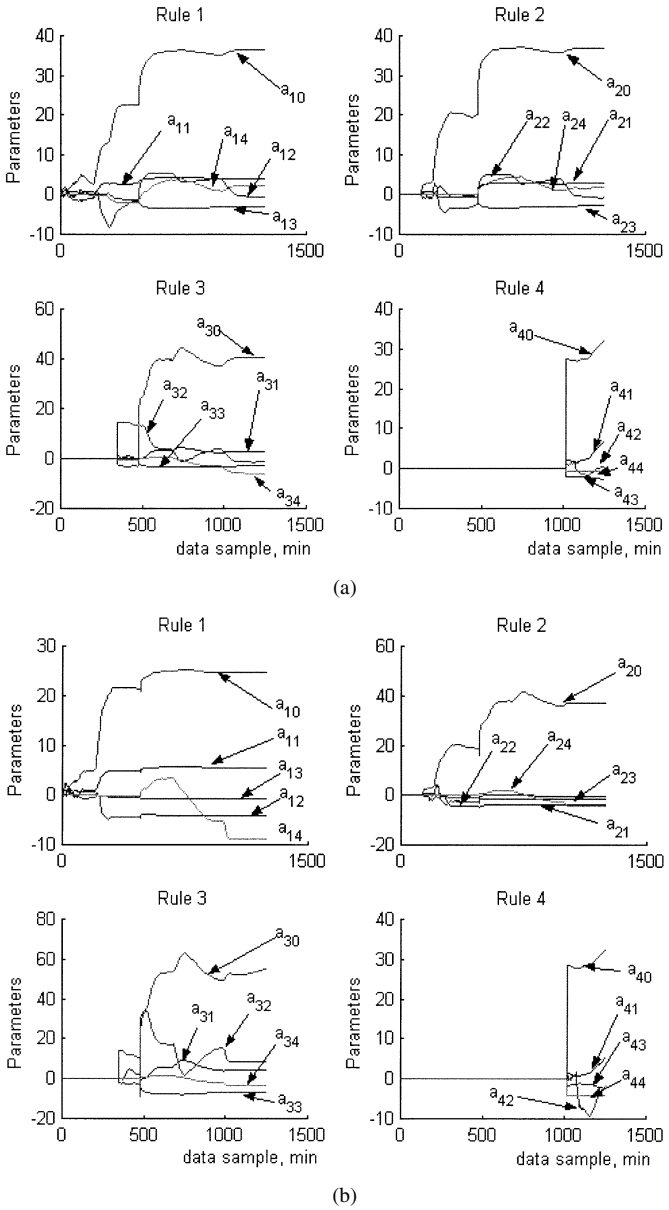

Fig. 11 Evolution of parameters of consequent part (global criteria).

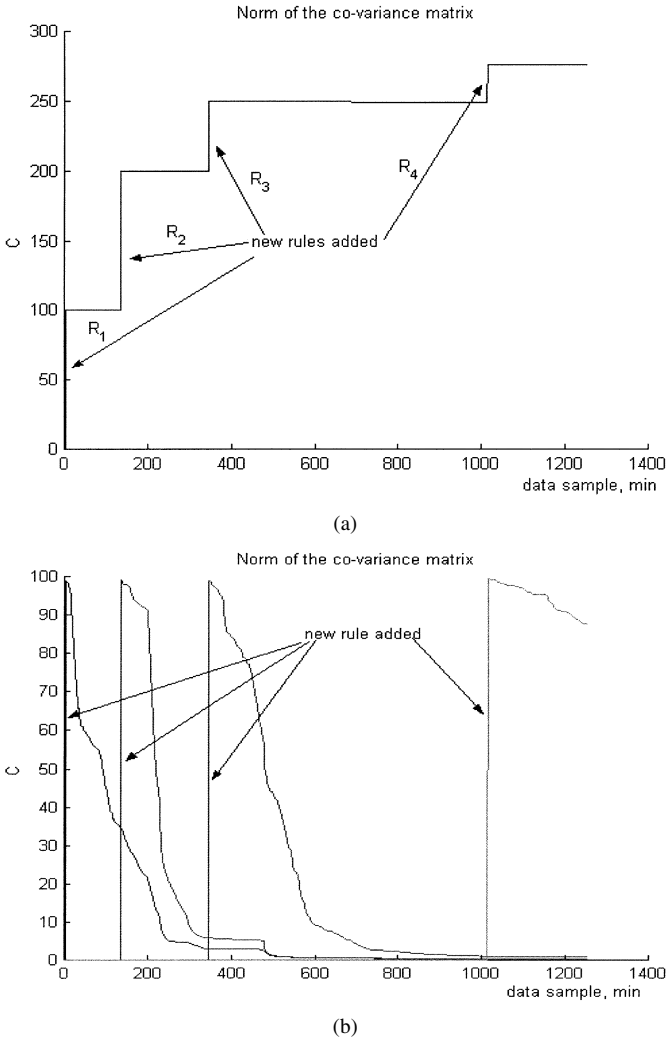

Fig. 12 (a) Evolution of the norm of the co-variance (global estimation). (b) Evolution of the norm of the co-variance (local estimation).

but using nonrecursive moving window [35] yields for the same problem $\mathrm{NDEI} = 0.004$ evolving to 35 rules.

In order to test the robustness of the eTS model a 5% random noise has been added to the standard Mackey–Glass time series. The eTS model has evolved to 124 rules with different centers and the $\mathrm{NDEI} = 0.309\,58$. From Fig. 13(d) and Fig. 13(f) it can be seen that initially the error is higher, but the TS model quickly up-grades its structure and reduces the error. Fig. 13(e) illustrates the evolution of the parameters of the first six fuzzy rules, which is similar to the case when no noise is considered in the data.

VI. Conclusion

An approach to *online* identification of ETS models is proposed in the paper. It is computationally effective, as it does not require re-training of the whole model. It is based on *recursive, noniterative* building of the rule base by unsupervised learning. The rule-based model evolves by replacement or upgrade of rules and parameter estimation.

The adaptive nature of this model in addition to the highly transparent and compact form of fuzzy rules makes them a promising candidate for *online* modeling and control of complex processes competitive to neural networks. The main advantages of the approach are:

TABLE III
Comparison of ETS with Other Evolving Models

Methods	Rules (nodes, units)	NDEI
DENFIS [31]	58 fuzzy rules	0.276
eTS model (this paper)	**113 fuzzy rules**	**0.0954**
RAN [32]	113 units	0.373
ESOM [33]	114 units	0.32
EFuNN [34]	193 rule nodes	0.401
DENFIS [31]	883 fuzzy rules	0.033
ESOM [33]	1000 units	0.044
Neural gas [32]	1000 units	0.062
EFuNN [34]	1125 rule nodes	0.094

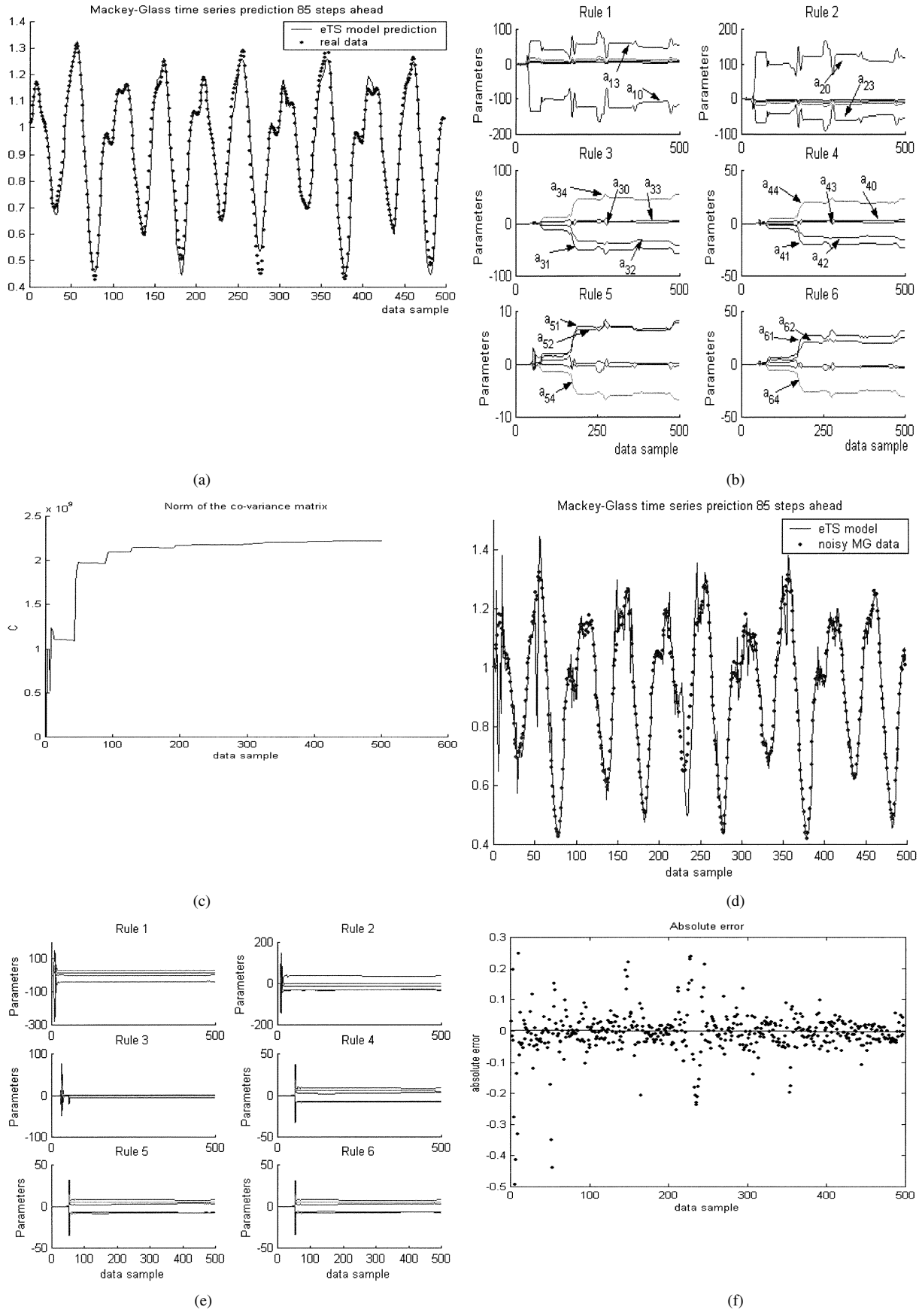

Fig. 13 (a) Prediction (85 steps ahead) of the Mackey-Glass chaotic time series by eTS model. (b) Evolution of parameters of linear sub-models for the first six fuzzy rules. (c) Evolution of the norm of the co-variance. (d) Prediction of the noisy Mackey-Glass chaotic time series by eTS model. (e) Evolution of parameters of linear sub-models for the first six fuzzy rules (noisy data). (f) Absolute error in prediction of the noisy Mackey-Glass chaotic time series by eTS model.

1) it can develop/evolve an existing model when the data pattern changes, while inheriting the rule base;
2) it can start to learn a process from a single data sample and improve the performance of the model predictions *online*;
3) it is *noniterative* and *recursive* and hence computationally very effective (the time necessary for calculation is a fraction of a second for a new data sample using a standard PC).

The proposed concept, has wide implications for many fields, including nonlinear adaptive control, fault detection and diagnostics, performance analysis of dynamical systems, time-series and forecasting, knowledge extraction, intelligent agents, behavior and modeling. The results illustrate the viability, efficiency and the potential of the approach when used with a limited amount of initial information, especially important in autonomous systems and robotics. Future implementation in various engineering problems is under consideration.

APPENDIX

A. WEIGHTED RECURSIVE LEAST SQUARES ALGORITHM

The wRLS algorithm could be derived from the weighted pseudo-inversion (16) expressing the matrices as sums and regrouping the components in a similar way as RLS is derived from LS [2], [30]

$$\hat{\pi}_{ik} = c_{ik}\left(\sum_{j=1}^{k-1} x_{ej}\lambda_1(x_j)y_j + x_{ek}\lambda_1(x_k)y_k\right) \tag{A1}$$

where $c_{ik} = \left(\sum_{j=1}^{k} x_{ej}^T\lambda_i(x_j)x_{ej}\right)^{-1}$ is the co-variance matrix.

Using this expression for the estimate based on $k-1$ data and regrouping we have

$$\begin{aligned}\sum_{j=1}^{k-1} x_{ej}\lambda_1(x_j)y_j &= c_{ik-1}^{-1}\hat{\pi}_{ik-1}\\ &= c_{ik}^{-1}\hat{\pi}_{ik-1} - x_{ek}\lambda_1(x_k)x_{ek}^T\hat{\pi}_{ik-1}.\end{aligned} \tag{A2}$$

Substituting (A2) in (A1) and using the matrix inversion Lemma (Lemma 3.1 from [30], p. 65) we arrive at

$$\begin{aligned}\hat{\pi}_{ik} &= \hat{\pi}_{ik-1} - c_{ik}x_{ek}\lambda_1(x_k)x_{ek}^T\hat{\pi}_{ik-1} + c_{ik}x_{ek}\lambda_1(x_k)y_k\\ &= \hat{\pi}_{ik-1} + c_{ik}x_{ek}\lambda_1(x_k)\left(y_k - x_{ek}^T\hat{\pi}_{ik-1}\right)\end{aligned} \tag{A3}$$

which is equivalent to (17). In a similar way, from the definition of the co-variance matrix for the estimation based on k data we have

$$\begin{aligned}c_{ik} &= \left(\sum_{j=1}^{k-1} x_{ej}^T\lambda_1(x_j)x_{ej} + x_{ek}^T\lambda_1(x_k)x_{ek}\right)^{-1}\\ &= \left(c_{k-1}^{-1} + \lambda_1(x_k)x_{ek}^Tx_{ek}\right)^{-1}\\ &= c_{ik-1} - c_{ik-1}\lambda_1(x_k)x_{ek}\\ &\quad\times\left(I + \lambda_1(x_k)x_{ek}^Tc_{ik-1}x_{ek}\right)^{-1}x_{ek}^Tc_{ik-1}\end{aligned} \tag{A4}$$

which is equivalent to (18).

B. RECURSIVE POTENTIALS CALCULATION

Starting form the formula of the potential (19) and expressing the projections of the distances in an explicit form for the time step k we have

$$P_k(z_k) = \frac{1}{1+\frac{1}{(k-1)}\sum_{i=1}^{k-1}\sum_{j=1}^{n+1}\left\{\left(z_k^j\right)^2 - 2z_k^jz_i^j + \left(z_i^j\right)^2\right\}}. \tag{A5}$$

Regrouping we have (A6) shown at the bottom of the page, which is equivalent to (20).

C. RECURSIVE UPDATE OF FOCAL POINT'S POTENTIAL

If a data point is accepted to be the focal point of a cluster/rule at time $k-1 (z_1^* := z_{k-1}$; for $k > 2)$ then its potential is calculated according to (19) as

$$P_{k-1}(z_i^*) = \frac{1}{1+\frac{1}{(k-2)}\sum_{i=1}^{k-2}\sum_{j=1}^{n+1}\left(d_{i(k-1)}^j\right)^2}. \tag{A7}$$

We can re-order expressing the sums explicitly

$$\sum_{i=1}^{k-2}\sum_{j=1}^{n+1}\left(d_{ik-1}^j\right)^2 = (k-2)\left(\frac{1}{P_{k-1}(z_i^*)} - 1\right). \tag{A8}$$

At the next time step (k) the potential have to be updated in order to accommodate the influence of the new data z_k on this center

$$\begin{aligned}&P_k(z_i^*)\\ &= \frac{1}{1+\frac{1}{(k-1)}\left(\sum_{i=1}^{k-2}\sum_{j=1}^{n+1}\left(d_{ik-1}^j\right)^2 + \sum_{j=1}^{n+1}\left(d_{k(k-1)}^j\right)^2\right)}.\end{aligned} \tag{A9}$$

By substituting (A8) into (A9) we have

$$\begin{aligned}&P_k(z_i^*)\\ &= \frac{1}{1+\frac{1}{(k-1)}\left((k-2)\left(\frac{1}{P_{k-1}(z_i^*)}-1\right) + \sum_{j=1}^{n+1}\left(d_{k(k-1)}^j\right)^2\right)}.\end{aligned} \tag{A10}$$

By regrouping, we arrive at (A11) shown at the top of the next page, which is equivalent to (21).

D. CO-VARIANCE MATRIX UPDATE

Let us introduce a vector of inputs that are weighted by the nonnormalized firing levels of the rules similarly to the notations used in (10) as

$$\varphi_k = \left[\tau_1(x_k)x_{ek}^T, \tau_2(x_k)x_{ek}^T, \ldots, \tau_R(x_k)x_{ek}^T\right]^T. \tag{A12}$$

$$P_k(z_k) = \frac{1}{1+\frac{1}{(k-1)}\left((k-1)\sum_{i=1}^{n+1}\left(z_k^j\right)^2 - 2\sum_{j=1}^{n+1}z_k^j\sum_{i=1}^{k-1}z_i^j + \sum_{i=1}^{k-1}\sum_{j=1}^{n+1}\left(z_i^j\right)^2\right)} \tag{A6}$$

$$P_k(z_i^*) = \frac{(k-1)P_{k-1}(z_i^*)}{kP_{k-1}(z_i^*) - P_{k-1}(z_i^*) + k - 2 - kP_{k-1}(z_i^*) + 2P_{k-1}(z_i^*) + P_{k-1}(z_i^*)\sum_{j=1}^{n+1}\left(d_{k(k-1)}^j\right)^2} \tag{A11}$$

Then it is obvious that

$$\psi_k = \frac{1}{\sum_{j=1}^{R}\tau_j}\varphi_k. \tag{A12a}$$

From the expression for the Kalman filter for the update of the co-variance matrices (12) we have

$$C_k = C_{k-1} - \frac{C_{k-1}\varphi_k\varphi_k^T C_{k-1}}{\left(\sum_{j=1}^{R}\tau_j\right)^2 + \varphi_k^T C_{k-1}\varphi_k} \tag{A13}$$

or expressing the history until time $\boldsymbol{k}$ in an explicit way

$$C_k = \Omega I - \sum_{i=1}^{k}\frac{A_i}{B_i + F} \tag{A14}$$

where $A_i = C_{i-1}\varphi_i\varphi_i^T C_{i-1}$; $B_i = \varphi_i^T C_{i-1}\varphi_i$; $F = \left(\sum_{j=1}^{R}\tau_j\right)^2$ Let us suppose that the rule added at the step k had been added from the beginning. Then the co-variance matrix at time k would be

$$\tilde{C}_k = \Omega I - \sum_{i=1}^{k}\frac{A_i}{B_i + \left(\sum_{j=1}^{R}\tau_j + \tau_{R+1}\right)}$$

or

$$\tilde{C}_k = \Omega I - \sum_{i=1}^{k}\frac{A_i}{B_i + F + \delta F_1 + \delta F_2} \tag{A15}$$

where $\delta F_1 = \tau_{R+1}^2$; $\delta F_2 = 2\tau_{R+1}\sum_{j=1}^{R}\tau_j$. It can be seen that adding a rule at time step k results in a corruption of the co-variance matrix, which is expressed in an increase of the denominator of the part subtracted from $C_0 = \Omega I$. It should be noted that the values of δF_1 and δF_2 are strongly less than 1 (because they are quadratic forms of membership functions). B_i could be a big number since it is a quadratic form of the input data multiplied by the co-variance matrix. F is bigger then δF_1 since it is a sum of R positive membership functions, while δF_1 is only one MF. F is also bigger than δF_2 if $\tau_{R+1} > (1/2)\sum_{j=1}^{R}\tau_j$. Therefore, the role of the addends would be more significant only if *all* values of x_{ie} (for *all* past time steps) tend to 0 or the co-variance matrix tends to zero. The practical tests with a number of functions illustrate that the corruption of the covariance matrix by the addition of a new rule is marginal.

We approximate this (normally small) influence by an inverse mean of average type of correction. The logic is following. From (A14), (A15) we have that the corrupted co-variance matrix $(\tilde{C}_{k+1})$ is a function of the original one (C_{k+1})

$$\tilde{C}_{k+1} = f(C_{k+1}). \tag{A16}$$

An approximation of the function f could be the inverse squared mean, since the role of the corruption will decrease with increase of R and this is a squared dependence

$$\tilde{C}_{k+1} = C_{k+1}\frac{R^2+1}{R^2} = C_{k+1}\left(1 + \frac{1}{R^2}\right) = \rho C_{k+1} \tag{A17}$$

which is equivalent to (25).

REFERENCES

[1] D. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, pp. 568–576, Nov. 1991.

[2] L. Ljung, *System Identification, Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[3] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. New York: Wiley, 1994.

[4] T. A. Johanson and R. Murray-Smith, "Operating regime approach to nonlinear modeling and control," in *Multiple Model Approaches to Modeling and Control*, R. Murray-Smith and T. A. Johanson, Eds. Hants, U.K.: Taylor Francis, pp. 3–72.

[5] J. S. R. Jang, "ANFIS: adaptive network-based fuzzy inference systems," *IEEE Trans. Syst., Man Cybern.*, vol. 23, pp. 665–685, May/June 1993.

[6] C. K. Chiang, H.-Y. Chung, and J. J. Lin, "A self-learning fuzzy logic controller using genetic algorithms with reinforcements," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 460–467, Aug. 1997.

[7] P. P. Angelov, V. I. Hanby, R. A. Buswell, and J. A. Wright, "Automatic generation of fuzzy rule-based models from data by genetic algorithms," in *Advances in Soft Computing*, R. John and R. Birkenhead, Eds. Heidelberg, Germany: Springer-Verlag, 2001, pp. 31–40.

[8] F. Hoffmann and G. Pfister, "Learning of a fuzzy control rule base using messy genetic algorithms," in *Studies in Fuzziness and Soft Computing*, F. Herrera and J.L. Verdegay, Eds. Heidelberg, Germany: Physica Verlag, 1996, vol. 8, pp. 279–305.

[9] P. P. Angelov, *Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Springer-Verlag, 2002.

[10] B. Carse, T. C. Fogarty, and A. Munro, "Evolving fuzzy rule-based controllers using GA," *Fuzzy Sets Syst.*, vol. 80, pp. 273–294, 1996.

[11] P. P. Angelov, V. I. Hanby, and J. A. Wright, "HVAC systems simulation: a self-structuring fuzzy rule-based approach," *Int. J. Architectural Sci.*, vol. 1, no. 1, pp. 49–58, 2000.

[12] K. Shimojima, T. Fukuda, and Y. Hasegawa, "Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm," *Fuzzy Sets Syst.*, vol. 71, pp. 295–309, 1995.

[13] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man Cybern.*, vol. 15, pp. 116–132, 1985.

[14] M. Sugeno and M. Yasukawa, "A fuzzy logic based approach ot qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 7–31, Feb. 1993.

[15] D. P. Filev, T. Larsson, and L. Ma, "Intelligent control for automotive manufacturing-rule based guided adaptation," in *Proc. IEEE Conf. IECON'00*, Nagoya, Japan, Oct. 2000, pp. 283–288.

[16] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intel. Fuzzy Syst.*, vol. 2, pp. 267–278, 1994.

[17] J. Bezdek, "Cluster validity with fuzzy sets," *J. Cybern.*, vol. 3, no. 3, pp. 58–71, 1974.

[18] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global and local learning," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 530–537, Nov. 1998.

[19] EUNITE: EUropean network on intelligent technologies for smart adaptive systems, p. 4, 2000.

[20] F.-J. Lin, C.-H. Lin, and P.-H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 751–759, Oct. 2001.

[21] H. R. Berenji, "A reinforcement learning-based architecture for fuzzy logic control," *Int. J. Approx. Reasoning*, vol. 6, pp. 267–292, 1992.
[22] G. G. Yen and P. Meesad, "An effective neuro-fuzzy paradigm for machinery condition health monitoring," in *Proc. IEEE Int. Joint Conf. IJCNN'99*, Washington, DC, 1999, pp. 1567–1572.
[23] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Control Decision Conf.*, San Diego, CA, 1979, pp. 761–766.
[24] F. Klawon and P. E. Klement, "Mathematical analysis of fuzzy classifiers," *Lect. Notes Comp. Sci.*, vol. 1280, pp. 359–370, 1997.
[25] R. R. Yager and D. P. Filev, "Learning of fuzzy rules by mountain clustering," in *Proc. SPIE Conf. Applicat. Fuzzy Logic Technol.*, Boston, MA, 1993, pp. 246–254.
[26] K. L. Anderson, G. L. Blackenship, and L. G. Lebow, "A rule-based adaptive PID controller," in *Proc. 27th IEEE CDC'88*, 1988, pp. 564–569.
[27] D. P. Filev, "Rule-base guided adaptation for mode detection in process control," in *Proc. Joint 9th IFSA World Congr./20th NAFIPS Annu. Conf.*, Vancouver, BC, Canada, July 2001, pp. 1068–1073.
[28] R. Babuska, "Fuzzy modeling and identification," Ph.D. thesis, Univ. of Delft, Delft, The Netherlands, 1996.
[29] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computat.*, vol. 1, pp. 281–294, 1989.
[30] K. J. Astroem and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.
[31] N. K. Kasabov and Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 144–154, Apr. 2002.
[32] J. Platt, "A resource allocation network for function interpolation," *Neural Computat.*, vol. 3, pp. 213–225, 1991.
[33] D. Deng and N. Kasabov, "Evolving self-organizing maps for online learning, data analysis and modeling," in *Proc. IJCNN'2000 Neural Networks, Neural Comput.: New Challenges Perspectives New Millennium*, vol. VI, S.-I. Amari, C. L. Giles, M. Gori, and V. Piuri, Eds., New York, NY, 2000, pp. 3–8.
[34] N. Kasabov, "Evolving fuzzy neural networks—algorithms, applications and biological motivation," in *Methodologies for the Conception, Design and Application of Soft Computing*, T. Yamakawa and G. Matsumoto, Eds, Singapore: World Scientific, 1998, pp. 271–274.
[35] P. Angelov and R. Buswell, "Identification of evolving fuzzy rule-based models," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 667–677, Oct. 2002.

Plamen P. Angelov (M'99) received the Ph.D. degree from the Bulgarian Academy of Science, Bulgaria, in 1993.

Since June 2003, he has been a Lecturer in the Department of Communications Systems, Lancaster University, Lancaster, U.K. He was a Research Fellow in Loughborough University, U.K., from 1998 to 2003. He has been Visiting Research Fellow in CESAME, Catholic University of Louvain, Belgium, in 1997, Hannover University and HKI, Jena, Germany, from 1995 to 1996. He authored the monograph *Evolving Rule based Models: A Tool for Design of Flexible Adaptive Systems* (Heidelberg, Germany: Springer, 2002). His research has been funded by the EC, ASHARE, Research Councils of UK (EPSRC), Germany (DAAD and DFG), Belgium, Italy (CNR), Bulgaria (NFSI). His research interests are in intelligent data processing, particularly in evolving rule-based models, self-organizing and autonomous systems, evolutionary algorithms, optimization and optimal control in a fuzzy environment.

Dr. Angelov has been in the program and organizing committees of several conferences, including IFSA-2003, GECCO-2002, RASC-2002, FUBEST'94 and '96, BioPS'94, '95, '97.

Dimitar P. Filev (M'95–SM'97) received the PhD. degree in electrical engineering from the Czech Technical University, Czechoslovakia, in 1979.

He is a Staff Technical Specialist and a Manager of the Knowledge Based Systems and Control Department with Advanced Manufacturing Technology Development, Ford Motor Company specializing in industrial intelligent systems and technologies for control, diagnostics and decision making. Prior to joining Ford, he was Professor of information systems and Senior Research Associate at the Machine Intelligence Institute, Iona College, and Associate Professor at the Bulgarian Academy of Sciences. He is conducting research in control theory and applications, modeling of complex systems, and intelligent modeling and control. He has published three books and over 150 articles in refereed journals and conference proceedings. He holds nine U.S. patents.

Dr. Filev received the 1995 Award for Excellence from MCB University Press and three Henry Ford Technology Awards. He is an Associate Editor of the IEEE TRANSACTIONS. ON FUZZY SYSTEMS, the *International Journal of General Systems*, and the *International Journal of Approximate Reasoning*.